Improving Statistical Machine Translation Performance by Training Data Selection and Optimization

Yajuan Lü, Jin Huang and Qun Liu
Key Laboratory of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Sciences
P.O. Box 2704, Beijing 100080, China
{lvyajuan, huangjin,liuqun}@ict.ac.cn

Abstract

Parallel corpus is an indispensable resource for translation model training in statistical machine translation (SMT). Instead of collecting more and more parallel training corpora, this paper aims to improve SMT performance by exploiting full potential of the existing parallel corpora. Two kinds of methods are proposed: offline data optimization and online model optimization. The offline method adapts the training data by redistributing the weight of each training sentence pairs. The online method adapts the translation model by redistributing the weight of each predefined submodels. Information retrieval model is used for the weighting scheme in both methods. Experimental results show that without using any additional resource, both methods can improve SMT performance significantly.

1 Introduction

Statistical machine translation relies heavily on the available training data. Typically, the more data is used to estimate the parameters of the translation model, the better it can approximate the true translation probabilities, which will obviously lead to a higher translation performance. However, large corpora are not easily available. The collected corpora are usually from very different areas. For example, the parallel corpora provided by LDC come from quite different domains, such as Hongkong laws, Hangkong Hansards and Hongkong news. This results in the problem that a translation system trained on data from a particular domain(e.g. Hongkong Hansards) will perform poorly when translating text from a different domain(e.g. news articles). Our experiments also show that simply putting all these domain specific corpora together will not always improve translation quality. From another aspect, larger amount of training data also requires larger computational resources. With the increasing of training data, the improvement of translation quality will become smaller and smaller. Therefore, while keeping collecting more and more parallel corpora, it is also important to seek effective ways of making better use of available parallel training data.

There are two cases when we train a SMT system. In one case, we know the target test set or target test domain, for example, when building a specific domain SMT system or when participating the NIST MT evaluation[1]. In the other case, we are unaware of any information of the testing data. This paper presents two methods to exploit full potential of the available parallel corpora in the two cases. For the first case, we try to optimize the training data offline to make it match the test data better in domain, topic and style, thus improving the translation performance. For the second case, we first divide the training data into several domains and train submodels for each domain. Then, in the translation process, we try to optimize the predefined models according to the online input source sentence. Information retrieval model is used for similar sentences retrieval in both methods. Our preliminary experiments show that both methods can improve SMT performance without using any additional data.

[1] http://www.nist.gov/speech/tests/mt/

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 343–350, Prague, June 2007.

The remainder of this paper is organized as follows: Section 2 describes the offline data selection and optimization method. Section 3 describes the online model optimization method. The evaluation and discussion are given in section 4. Related work is introduced before concluding.

2 Offline training data optimization

In offline training data optimization, we assume that the target test data or target test domain is known before building the translation model. We first select sentences similar to the test text using information retrieval method to construct a small and adapted training data. Then the extracted similar subset is used to optimize the distribution of the whole training data. The adapted and the optimized training data will be used to train new translation models.

2.1 Similar data selection using TF-IDF

We use information retrieval method for similar data retrieval. The standard TF-IDF (Term Frequency and Inverse Document Frequency) term weighting scheme is used to measure the similarity between the test sentence and the training sentence.

TF-IDF is a similarity measure widely used in information retrieval. Each document D_i is represented as a vector $(w_{i1}, w_{i2}, ..., w_{in})$, n is the size of the vocabulary. w_{ij} is calculate as follows:

$$w_{ij} = tf_{ij} \times \log(idf_j)$$

where,

tf_{ij} is the term frequency(TF) of the *j*-th word in the vocabulary in the document D_i , i.e. the number of occurrences;

idf_j is the inverse document frequency(IDF) of the *j*-th word calculated as below:

$$idf_j = \frac{\# documents}{\# documents\ containing\ \text{j - th term}}.$$

The similarity between two documents is then defined as the cosine of the angle between the two vectors.

We perform information retrieval using the Lemur toolkit[2]. The source language part of the parallel training data is used as the document collection. Each sentence represents one document. Each sentence from the test data or test domain is used as one separate query. In the sentence retrieval process, both the query and the document are converted into vectors by assigning a term weight to each word. Then the cosine similarity is calculated proportional to the inner product of the two vectors. All retrieved sentences are ranked according to their similarity with the query. We pair each of the retrieved sentences with the corresponding target part and the top N most similar sentences pairs are put together to form an adapted parallel data. N ranges from one to several thousand in our experiments. Since Lemur toolkit gives the similarity score for each retrieved sentences, it is also possible to select the most similar sentences according to the similarity score.

Note that the selected similar data can contain duplicate sentences as the top N retrieval results for different test sentences can contain the same training sentences. The duplicate sentences will force the translation probability towards the more often seen words. Intuitively, this could help. In experiment section, we will compare experimental results by keeping or removing duplicates to see how the duplicate sentences affect the translations.

The selected subset contains the similar sentences with the test data or test domain. It matches the test data better in domain, topic and style. Hopefully, training translation model using this adapted parallel data may helpful for improving translation performance. In addition, the translation model trained using the selected subset is usually much smaller than that trained using the whole translation data. Limiting the size of translation model is very important for some real applications. Since SMT systems usually require large computation resource. The complexity of standard training and decoding algorithm depends mainly on the size of the parallel training data and the size of the translation model. Limiting the size of the training data with the similar translation performance would also reduce the memories and speed up the translations.

In the information retrieval process, we only use the source language part for document indexing and query generating. It is easy to get source part of the test data. This is different from the common language model adaptation methods, which have to do at lease one pass machine translation to get the candidate English translation as query(Zhao 2004, Zhang 2006). So our method has the advantage that it is independent from the quality of baseline translation system.

[2] http://www.cs.cmu.edu/~lemur/

2.2 Training data optimization

There are two factors on training data that influence the translation performance of SMT system: the scale and the quality. In some sense, we improve the quality of the training data by selecting the similar sentence to form an adapted training set. However, we also reduce the scale of the training data at the same time. Although this is helpful for some small device applications, it is also possible to induce the data sparseness problem. Here, we introduce a method to optimize between the scale and the quality of the training data.

The basic idea is that we still use all the available training data; by redistributing the weight of each sentence pairs we adapt the whole training data to the test domain. In our experiments, we simply combine the selected small similar subset and the whole training data. The weights of each sentence pairs are changed accordingly. Figure 1 shows the procedure of the optimization.

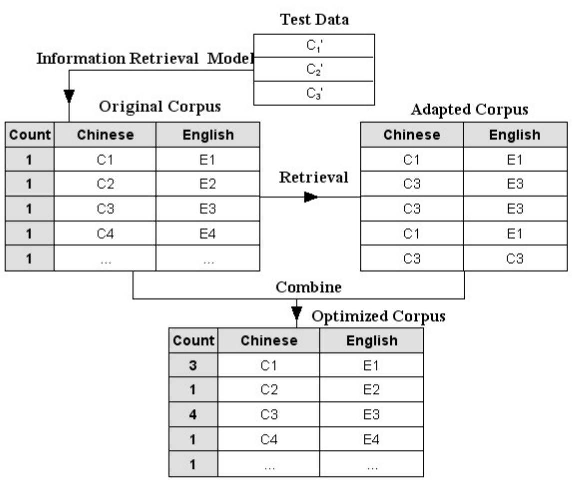

Figure 1. Training data optimization

As can be seen, through the optimization, the weight of the similar sentence pairs are increased, while the general sentence pairs still have an ordinary weight. This make the translation model inclined to give higher probabilities to the adapted words, and at the same time avoid the data sparseness problem. Since we only change the weight of the sentence pairs, and no new training data is introduced, the translation model size trained on the optimized data will keep as the same as the original one. We use GIZA++ toolkit[3] for word alignment training in the training process. The input training file formats for GIZA++ is as follows: Each training sentence pair is stored in three lines. The first line is the number of times this sentence pair occurred. The second line is the source sentence where each token is replaced by its unique integer id and the third is the target sentence in the same format. To deal with our optimized training data, we only need to change the number of sentence pairs in the first line accordingly. This will not call for extra training time and memory for the whole training process.

It might be beneficial to investigate other sophisticated weighting schemes under the similar idea, such as to give more precise fractional weights to the sentences according the retrieval similarity scores.

3 Online model optimization

In most circumstances, we don't know exactly the test data or the test domain when we train a machine translation system. This results in the fact that the performance of the translation system highly depends on the training data and the test data it is used in. To alleviate this blindfold status and maximize the potential of the available training corpora, we propose a novel online model optimization method.

The basic idea is that: several candidate translation models are prepared in training stage. In particularly, a general model is also prepared. Then, in the translation process, the similarity between the input sentence and the predefined models is calculated online to get the weights of each model. The optimized model is used to translate the input sentence.

There are two problems in the method: how to prepare submodels in training process and how to optimize the model weight online in translation process.

3.1 Prepare the submodels

There are several ways to prepare submodels in training process. If the training data comes from very different sources, we can divide the data according to its origins. Otherwise, we can use clustering method to separate the training corpus into several classes. In addition, our offline data adaptation method can also be used for submodel preparation. For each candidate domain, we can use the

[3] http://www.fjoch.com/GIZA++.html

source side of a small corpus as queries to extract a domain specific training set. In this case, a sentence pair in the training data may occur in several sub training data, but this doesn't matter. The general model is used when the online input is not similar to any prepared submodels. We can use all available training data to train the general model since generally larger data can get better model even there are some noises.

3.2 Online model weighting

We also use TF-IDF information retrieval method for online model weighting. The procedure is as follows:

For each input sentence:

1. *Do IR on training data collection, using the input sentence as query.*
2. *Determine the weights of submodels according to the retrieved sentences.*
3. *Use the optimized model to translate the sentence.*

The information retrieval process is the same as the offline data selection except that each retrieved sentence is attached with the sub-corpus information, i.e. it belongs to which sub-models in the training process.

With the sub-corpus information, we can calculate the weights of submodels. We get the top N most similar sentences, and then calculate proportions of each submodel's sentences. The proportion can be calculated use the count of the sentences or the similarity score of the sentences. The weight of each submodel can be determined according to the proportions.

Our optimized model is the log linear interpolation of the sub-models as follows:

$$\hat{p}(e \mid c) = p_0(e \mid c)^{\delta_0} \times \prod_{i=1}^{M} p_i(e \mid c)^{\delta_i}$$

$$\hat{e} = \arg\max_e (\delta_0 \log(p_0(e \mid c)) + \sum_{i=1}^{M} \delta_i \log(p_i(e \mid c)))$$

where, p_0 is the probability of general model, p_i is the probability of submodel *i*. δ_0 is the weight of general model. δ_i is the weight of submodel *i*. Each model *i* is also implemented using log linear model in our SMT system. So after the log operation, the submodels are interpolated linearly.

In our experiments, the interpolation factor δ_i is determined using the following four simple weighting schemes:

Weighting scheme 1:

$$\delta_0 = 0; \quad \delta_{max\_model} = 1; \quad \delta_{i \neq max\_model} = 0;$$

Weighting scheme 2:

if Proportion(*max_model*) > 0.5
  Use weighting scheme1;
else
  $\delta_0 = 1; \quad \delta_i = 0;$

Weighting scheme 3:

$\delta_0 = 0;$

$\delta_i = \text{Proportion}\ (model_i);$

Weighting scheme 4:

if Proportion(*max_model*) > 0.5
  Use weighting scheme3;
else
  $\delta_0 = 0.5;$
  $\delta_i = 0.5 \times \text{Proportion}\ (model_i);$

where, $model_i$ is the *i*-th submodel, $i = (1 \ldots M)$. Proportion ($model_i$) is the proportion of $model_i$ in the retrieved results. We use count for proportion calculation. *max_model* is the submodel with the max proportion score.

The training and translation procedure of online model optimization is illustrated in Figure 2.

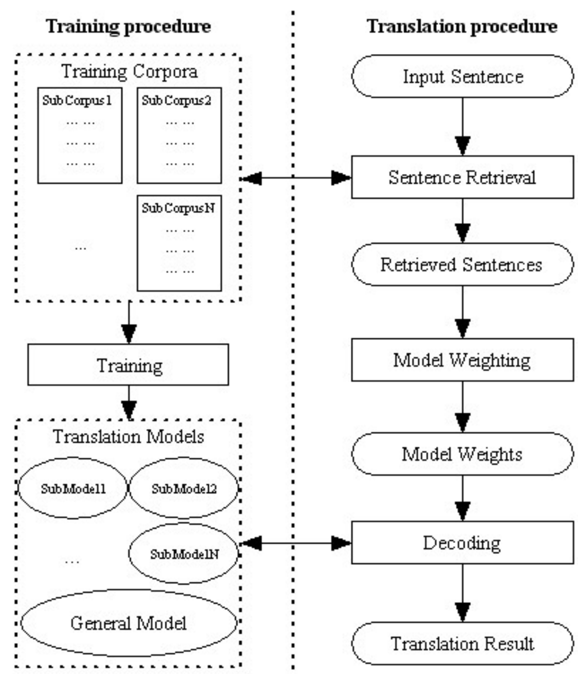

Figure 2. Online model optimization

The online model optimization method makes it possible to select suitable models for each individual test sentence. Since the IR process is done on a fixed training data, the size of the index data is quite small compared with the web IR. The IR process will not take much time in the translation.

4 Experiments and evaluation

4.1 Experimental setting

We conduct our experiments on Chinese-to-English translation tasks. The baseline system is a variant of the phrase-base SMT system, implemented using log-linear translation model (He et al. 2006). The baseline SMT system is used in all experiments. The only difference between them is that they are trained on different parallel training data.

In training process, we use GIZA++[4] toolkit for word alignment in both translation directions, and apply "grow-diag-final" method to refine it (Koehn et al., 2003). We change the preprocess part of GIZA++ toolkit to make it accept the weighted training data. Then we use the same criterion as suggested in (Zens et al., 2002) to do phrase extraction. For the log-linear model training, we take minimum-error-rate training method as described in (Och, 2003). The language model is trained using Xinhua portion of Gigaword with about 190M words. SRI Language Modeling toolkit[5] is used to train a 4-gram model with modified Kneser-Ney smoothing(Chen and Goodman, 1998). All experiments use the same language model. This ensures that any differences in performance are caused only by differences in the parallel training data.

Our training data are from three LDC corpora as shown in Table 1. We random select 200,000 sentence pairs from each corpus and combine them together as the baseline corpus, which includes 16M Chinese words and 19M English words in total. This is the usual case when we train a SMT system, i.e. we simply combine all corpora from different origins to get a larger training corpus.

We use the 2002 NIST MT evaluation test data as our development set, and the 2005 NIST MT test data as the test set in offline data optimization experiments. In both data, each sentence has four human translations as references. The translation quality is evaluated by BLEU metric (Papineni et al., 2002), as calculated by mteval-v11b.pl[6] with case-sensitive matching of n-grams.

Corpus	LDC No.	Description	# sent. pairs
FBIS	LDC2003E14	FBIS Multilanguage Texts	200000
HK_Hansards	LDC2004T08	Hong Kong Hansards Text	200000
HK_News	LDC2004T08	Hong Kong News Text	200000
Baseline	-	All above data	600000

Table 1. Training corpora

4.2 Baseline experiments

We first train translation models on each sub training corpus and the baseline corpus. The development set is used to tune the feature weights. The results on test set are shown in Table 2.

System	BLEU on dev set	BLEU on test set
FBIS	0.2614	0.2331
HK_Hansards	0.1679	0.1624
HK_News	0.1748	0.1608
Baseline	0.2565	0.2363

Table 2. Baseline results

From the results we can see that although the size of each sub training corpus is similar, the translation results from the corresponding system are quite different on the same test set. It seems that the FBIS corpus is much similar to the test set than the other two corpora. In fact, it is the case. The FBIS contains text mainly from China mainland news stories, while the 2005 NIST test set also include lots of China news text. The results illustrate the importance of selecting suitable training data.

When combining all the sub corpora together, the baseline system gets a little better result than the sub systems. This indicates that larger data is useful even it includes some noise data. However, compared with the FBIS corpus, the baseline corpus contains three times larger data, while the improvement of translation result is not significant. This indicates that simply putting different corpora together is not a good way to make use of the available corpora.

[4] http://www.fjoch.com/GIZA++.html

[5] http://www.speech.sri.com/projects/srilm/

[6] http://www.nist.gov/speech/tests/mt/resources/scoring.htm

4.3 Offline data optimization experiments

We use baseline corpus as initial training corpus, and take Lemur toolkit to build document index on Chinese part of the corpus. The Chinese sentences in development set and test set are used as queries. For each query, N = 100, 200, 500, 1000, 2000 similar sentences are retrieved from the indexed collection. The extracted similar sentence pairs are used to train the new adapted translation models. Table 3 illustrates the results. We give the distinct pair numbers for each adapted set and compare the size of the translation models. To illustrate the effect of duplicate sentences, we also give the results with duplicates and without duplicates (distinct).

System	Distinct pairs	Size of trans model	BLEU on duplicates	BLEU on distinct
Baseline	600000	2.41G	0.2363	0.2363
Top100	91804	0.43G	0.2306	0.2346
Top200	150619	0.73G	0.2360	0.2345
Top500	261003	1.28G	0.2415	0.2370
Top1000	357337	1.74G	0.2463	0.2376
Top2000	445890	2.11G	0.2351	0.2346

Table 3. Offline data adaptation results

The results show that:

1. By using similar data selection, it is possible to use much smaller training data to get comparable or even better results than the baseline system. When N=200, using only 1/4 of the training data and 1/3 of the model size, the adapted translation model achieves comparable result with the baseline model. When N=500, the adapted model outperforms the baseline model with much less training data. The results indicate that relevant data is better data. The method is particular useful for SMT applications on small device.

2. In general, using duplicate data achieves better results than using distinct data. This justifies our idea that give a higher weight to more similar data will benefit.

3. With the increase of training data size, the translation performance tends to improve also. However, when the size of corpus achieves a certain scale, the performance may drop. This maybe because that with the increase of the data, noisy data may also be included. More and more included noises may destroy the data. It is necessary to use a development set to determine an optimal size of N.

We combine each adapted data with the baseline corpus to get the optimized models. The results are shown in Table 4. We also compare the adapted models (TopN) and the optimized models (TopN+) in the table.

Without using any additional data, the optimized models achieve significant better results than the baseline model by redistributing the weight of training sentences. The optimized models also outperform adapted models when the size of the adapted data is small since they make use of all the available data which decrease the influence of data sparseness. However, with the increase of the adapted data, the performance of optimized models is similar to that of the adapted models.

System	Distinct pairs	BLEU on TopN	BLEU on TopN+
Baseline	600000	0.2363	0.2363
Top100+	600000	0.2306	0.2387
Top200+	600000	0.2360	0.2443
Top500+	600000	0.2415	0.2461
Top1000+	600000	0.2463	0.2431
Top2000+	600000	0.2351	0.2355

Table 4. Offline data optimization results

4.4 Online model optimization experiments

Since 2005 NIST MT test data tends bias to FBIS corpus too much, we build a new test set to evaluate the online model optimization method. We randomly select 500 sentences from extra part of FBIS, HK_Hansards and HK_News corpus respectively (i.e the selected 1500 test sentences are not included in any of the training set). The corresponding English part is used as translation reference. Note that there is only one reference for each test sentence. We also include top 500 sentence and their first reference translation of 2005 NIST MT test data in the new test set. So in total, the new test contains 2000 test sentences with one translation reference for each sentence. The test set is used to simulate SMT system's online inputs which may come from various domains.

The baseline translation results are shown in Table 5. We also give results on each sub test set (denotes as Xcorpus_part). Please note that the absolute BLEU scores are not comparable to the previous experiments since there is only one reference in this test set.

As expected, using the same domain data for training and testing achieves the best results as indicate by **bold fonts**. The results demonstrate again that relevant data is better data.

To test our online model optimization method, we divide the baseline corpus according to the origins of sub corpus. That is, the FBIS, HK_ Hansards and HK_News models are used as three sub-models and the baseline model is used as general model. The four weighting schemes described in section 3.2 are used as online weighting schemes individually. The experimental results are shown in Table 6. S_*i* indicates the system using weighting scheme *i*.

System / Test data	FBIS	HK_ Hansards	HK_ News	Baseline
FBIS-part	**0.1096**	0.0687	0.0622	0.1030
HK_Hans_part	0.0726	**0.0918**	0.0846	0.0897
HK_News_part	0.0664	0.0801	**0.0936**	0.0870
MT05_part	0.1130	0.0805	0.0776	0.1116
Whole test set	0.0937	0.0799	0.0781	0.0993

Table 5. Baseline results on new test set

System / Test data	S_1	S_2	S_3	S_4
FBIS-part	0.1090	0.1090	0.1089	0.1089
HK_Hans_part	0.0906	0.0903	0.0902	0.0902
HK_News_part	0.0952	0.0950	0.0933	0.0934
MT05_part	0.1119	0.1123	0.1149	0.1151
Whole test set	0.1034	0.1034	0.1038	0.1038

Table 6. Online model optimization results

Different weighting schemes don't show significant improvements from each other. However, all the four weighting schemes achieve better results than the baseline system. The improvements are shown not only on the whole test set but also on each part of the sub test set. The results justify the effectiveness of our online model optimization method.

5 Related work

Most previous research on SMT training data is focused on parallel data collection. Some work tries to acquire parallel sentences from web (Nie et al. 1999; Resnik and Smith 2003; Chen et al. 2004). Others extract parallel sentences from comparable or non-parallel corpora (Munteanu and Marcu 2005, 2006). These work aims to collect more parallel training corpora, while our work aims to make better use of existing parallel corpora.

Some research has been conducted on parallel data selection and adaptation. Eck et al. (2005) propose a method to select more informative sentences based on n-gram coverage. They use n-grams to estimate the importance of a sentence. The more previously unseen n-grams in the sentence the more important the sentence is. TF-IDF weighting scheme is also tried in their method, but didn't show improvements over n-grams. This method is independent of test data. Their goal is to decrease the amount of training data to make SMT system adaptable to small devices. Similar to our work, Hildebrand et al. (2005) also use information retrieval method for translation model adaptation. They select sentences similar to the test set from available in-of-domain and out-of-domain training data to form an adapted translation model. Different from their work, our method further use the small adapted data to optimize the distribution of the whole training data. It takes the full advantage of larger data and adapted data. In addition, we also propose an online translation model optimization method, which make it possible to select adapted translation model for each individual sentence.

Since large scale monolingual corpora are easier to obtain than parallel corpora. There has some research on language model adaptation recent years. Zhao et al. (2004) and Eck et al.(2004) introduce information retrieval method for language model adaptation. Zhang et al.(2006) and Mauser et al.(2006) use adapted language model for SMT re-ranking. Since language model is built for target language in SMT, one pass translation is usually needed to generate n-best translation candidates in language model adaptation. Translation model adaptation doesn't need a pre-translation procedure. Comparatively, it is more direct. Language model adaptation and translation model adaptation are good complement to each other. It is possible that combine these two adaptation approaches could further improve machine translation performance.

6 Conclusion and future work

This paper presents two new methods to improve statistical machine translation performance by making better use of the available parallel training corpora. The offline data selection method adapts the training corpora to the test domain by retrieving similar sentence pairs and redistributing their weight in the training data. Experimental results show that the selected small subset achieves comparable or even better performance than the baseline system with much less training data. The optimized training data can further improve translation performance without using any additional resource. The online model optimization method adapts the translation model to the online test sentence by redistributing the weight of each predefined submodels. Preliminary results show the effectiveness of the method. Our work also demonstrates that in addition to larger training data, more relevant training data is also important for SMT model training.

In future work, we will improve our methods in several aspects. Currently, the similar sentence retrieval model and the weighting schemes are very simple. It might work better by trying other sophisticated similarity measure models or using some optimization algorithms to determine submodel's weights. Introducing language model optimization into our system might further improve translation performance.

Acknowledgement

This work was supported by National Natural Science Foundation of China, Contract No. 60603095 and 60573188.

References

Jisong Chen, Rowena Chau, Chung-Hsing Yeh 2004. *Discovering Parallel Text from the World Wide Web*. ACSW Frontiers 2004: 157-161

Stanley F. Chen and Joshua Goodman. 1998. *An Empirical Study of Smoothing Techniques for Language Modeling*. Technical Report TR-10-98, Harvard University Center for Research in Computing Technology.

Matthias Eck, Stephan Vogel, and Alex Waibel 2004. *Language Model Adaptation for Statistical Machine Translation Based on Information Retrieval*. Proceedings of Fourth International Conference on Language Resources and Evaluation:327-330

Matthias Eck, Stephan Vogel, Alex Waibel 2005. *Low cost portability for statistical machine translation based on n-gram coverage*. MT Summit X: 227-234.

Zhongjun He, Yang Liu, Deyi Xiong, Hongxu Hou, and Qun Liu 2006. *ICT System Description for the 2006 TC-STAR Run#2 SLT Evaluation*. Proceedings of TC-STAR Workshop on Speech-to-Speech Translation: 63-68

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. *Statistical phrase-based translation*. Proceedings of HLT-NAACL 2003: 127–133.

Arne Mauser, Richard Zens, Evgeny Matusov, Sasa Hasan, Hermann Ney 2006. *The RWTH Statistical Machine Translation System for the IWSLT 2006 Evaluation*. Proceedings of International Workshop on Spoken Language Translation.:103-110

Dragos Stefan Munteanu and Daniel Marcu 2005. *Improving Machine Translation Performance by Exploiting Comparable Corpora*. Computational Linguistics, 31 (4): 477-504

Dragos Stefan Munteanu and Daniel Marcu 2006. *Extracting Parallel Sub-Sentential Fragments from Comparable Corpora*. ACL-2006: 81-88

Jian-Yun Nie, Michel Simard, Pierre Isabelle, Richard Durand 1999. *Cross-Language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Texts in the Web.* SIGIR-1999: 74-81

Franz Josef Och 2003. *Minimum Error Rate Training in Statistical Machine Translation.* ACL-2003:160-167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *Bleu: a Method for Automatic Evaluation of Machine Translation*. ACL-2002: 311–318

Philip Resnik and Noah A. Smith 2003. *The Web as a Parallel Corpus.* Computational Linguistics 29(3): 349-380

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel 2005. *Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval*. Proceedings of EAMT 2005: 133-142.

Richard Zens, Franz Josef Och, Hermann Ney 2002. *Phrase-Based Statistical Machine Translation*. Annual German Conference on AI, KI 2002, Vol. LNAI 2479: 18-32

Ying Zhang, Almut Silja Hildebrand, Stephan Vogel 2006. *Distributed Language Modeling for N-best List Re-ranking.* EMNLP-2006:216-223

Bing Zhao, Matthias Eck, Stephan Vogel 2004. *Language Model Adaptation for Statistical Machine Translation with structured query models*. COLING-2004